**基于用户和电影特征信息的电影推荐系统**

**功能描述：**

初步计划实现以下几个功能：

1. 无用户登录时按照评分人数，评分高低等指标进行首页的推荐。选取评分人数最多的50部电影中平均评分最高的5部作为高分电影推荐。选取某一日期（一般为当月月初）之后上映的电影中评分最高的的5部作为最新电影推荐（若不足5部则全部推荐）。
2. 有用户登录时按照用户近期的评分记录以及电影间的相似性进行首页电影的推荐。用户登录后需要刷新首页的推荐名单，**增加**和用户相关的电影推荐。

用户已评分过的电影认为其已观看，推荐时应避免。选取用户评分最高的5部电影，分别计算与其最相似的5部（共25部），选取其中不重复且未评分的电影电影进行推荐。

*可能增加的功能：选取用户最近10次不同字段的搜索记录中搜索次数最多的5个，根据这些字段在索引表中找到对应的电影，随机选取5个进行推荐。*

1. 用户点击进入某一电影项后对相关电影进行推荐。寻找与该电影最相似的10部电影进行推荐。

**技术框架：**

需建立用户-评分表，电影间相似度表。

计划使用Pyspark的TF-IDF计算两种方式得到电影相似度（基于word2vec的计算由同组同学完成）。

Pyspark是Spark为Python提供的API，用于处理大规模数据。其主要特点就是提供了一个集群的分布式抽象RDD，并由RDD的转换操作组成DAG，DAG最终作为一个task提交给Spark执行。

TF-IDF是表示文本特征的一种方式，其主要思想为，如果某个词或短语在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。计算某个单词在文档d中的词频以及整个corpus中的逆文档频率（用于词语的重要性），两者的乘积称为某个单词的TF-IDF值，文档可以被表述为相应的TF-IDF向量，用于计算文本相似度。

推荐算法比较简单，此处不再赘述。

**进度安排：**10.28 – 10.30 方法调研

10.31-11.8 完成推荐模块的编写

11.9 – 11.10 联调

11.11-11.13 撰写推荐部分算法的说明文档